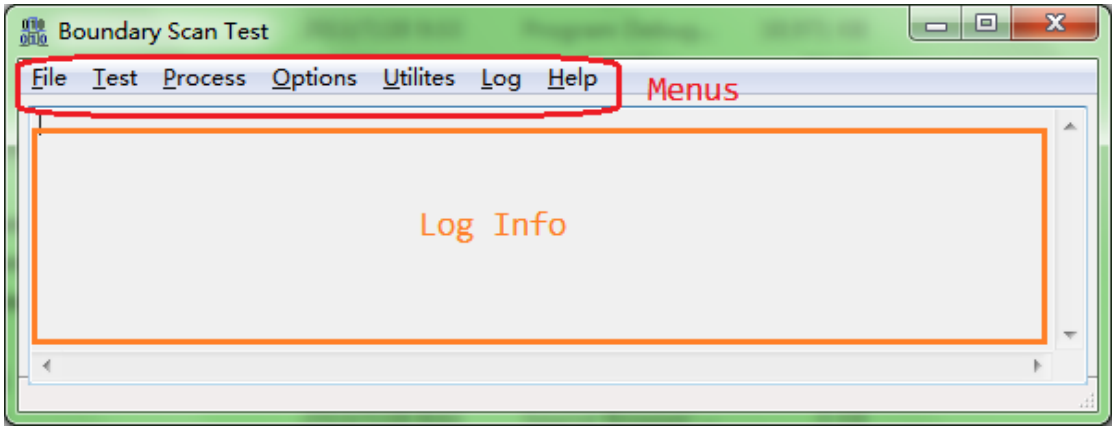
Contents

Main User Interface 3
Before Start 3
Set TCK Frequency 5
Manual Test 6
Preparation 6
If you have BSDL file for every device 6
If you don't get BSDL file for every device 7
Prefix Instruction Length 8
Post Instruction Length 8
Setting 8
Operation 9
I/O 9
Utility for Setting I/O 9
Utilities 11
Lookup BSC No. by Pin Location 11
Lookup BSC No. by Port Name 12
Output Setting 12
Input Setting 13
More Devices... 13
Test 13
Single Device Test 14
Preparation 14
Output Setting 16
Expected Input 17
Save 17
Utilities 17
Edit Saved Setting File 18
Steps 18
Example 1 20
Test on Good Board 20
Short-Circuit Issue 20
More... 21
Example 2 21
Test on Good Board 22
Short Circuit with Power 22
Short Circuit with GND 23
Short-Circuit Between Pins 23
More... 24
Inter-Device Test 24
Steps 24
About the Parameters 25
An Example 26
Tips 26

Run the Test 26
UI Difference Under Different Launching Mode 28
PLD Configuration 29
Get .svf File 29
Start 29
Set .svf File 29
Run 31

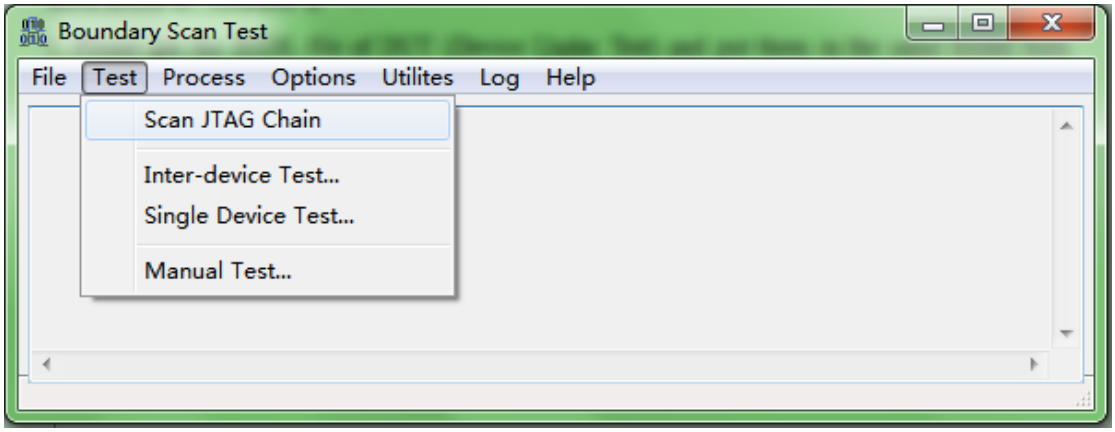
Note: There may be some changes between versions. So, your software may look a bit different from this manual.

Main User Interface

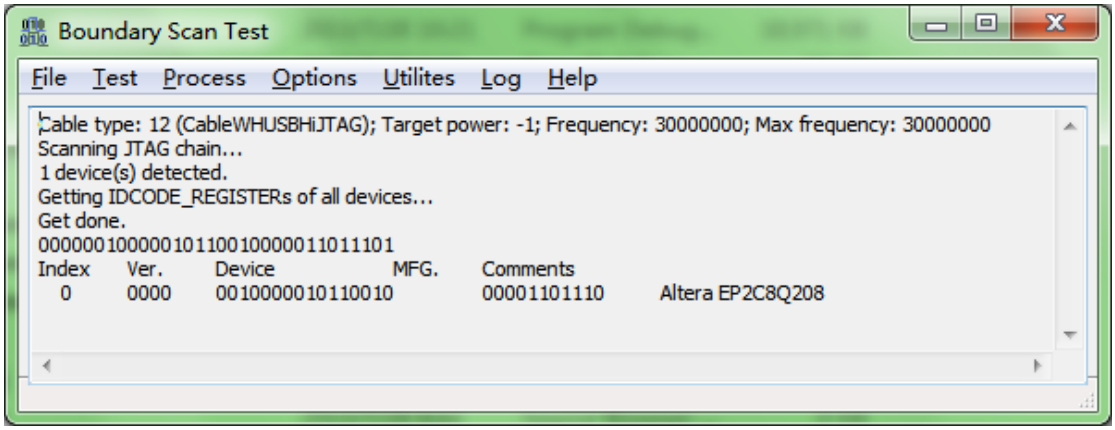


Before Start

1, Please get the BSDL file of DUT (Device Under Test) and put them in the same folder with software. If there multi devices in JTAG chain, please try to get all BSDL files. And if you could not get every BSDL file, you must know each device's JTAG instruction length.
2, Power off the target board.
3, Connect the cables between computer and target board.
4, Power up the target board;
5, Click to run the software.
6, When you see the main UI dialog, please select menu **Test - Scan JTAG Chain** to check how many devices detected by the software. See screenshot below:



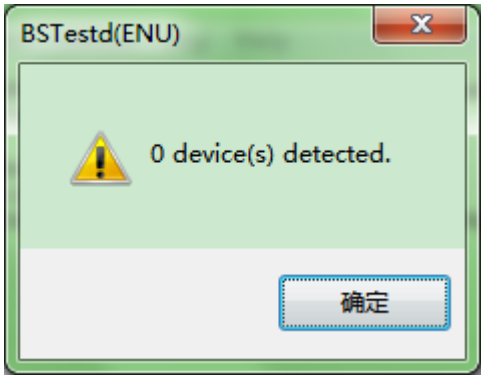
If any device is detected, device ID and other info will be displayed. See screenshot below:


Boundary Scan Test
File Test Process Options Utilites Log Help
Cable type: 12 (CableWHUSBHiJTAG); Target power: -1; Frequency: 30000000; Max frequency: 30000000
Scanning JTAG chain...
1 device(s) detected.
Getting IDCODE_REGISTERs of all devices...
Get done.
00000010000010110010000011011101
Index Ver. Device MFG. Comments
0 0000 0010000010110010 00001101110 Altera EP2C8Q208

Another log example of Xilinx ML505 reference design board.

```
Cable type: 10; Target power: -1; Frequency: 6000000
Scanning JTAG chain...
5 device(s) detected.
Getting IDCODE_REGISTERs of all devices...
Get done.
11110101000001011001000010010011
11110101000001011001000010010011
01011001011000001000000010010011
00001010000000000000001000010010011
11000010101010010110000010010011
Index  Ver.   Device           MFG.       Comments
   0   1111   0101000001011001 00001001001   Xilinx XCF32P
   1   1111   0101000001011001 00001001001   Xilinx XCF32P
   2   0101   1001011000001000 00001001001   Xilinx XC95144XL
   3   0000   1010000000000001 00001001001   Xilinx XCCACE
   4   1100   0010101010010110 00001001001   Xilinx XC5LX50T
```

If no device detected, a message box is popped, and please refer to troubleshooting below.



If no device detected, please check:

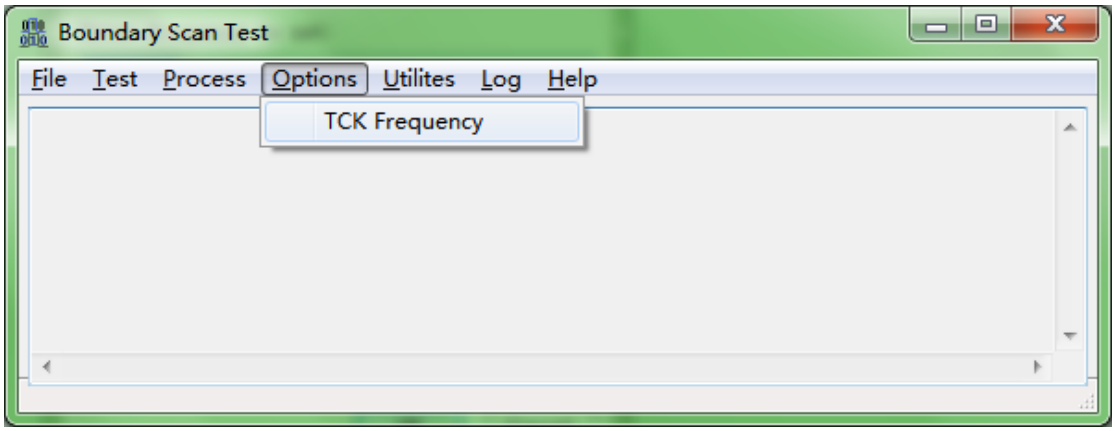
- Cable is connected correctly;
- JTAG circuit of target board works;
- Target board is powered OK;

If the device list doesn't match with the board, please read target board's manual to confirm jumper settings or configurations. If you are using WH-USB-HiJTAG cable, you could try to slow the TCK frequency, please refer to Set TCK Frequency.

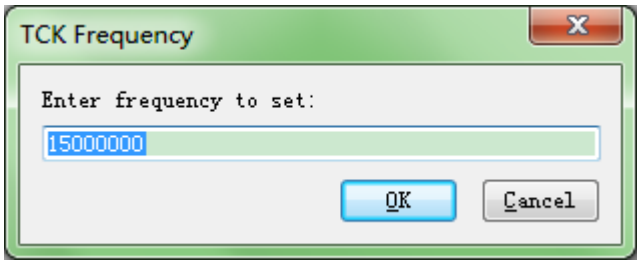
Only when device list detected by the software matches with the board, you can do any test with the software.

Set TCK Frequency

Select menu **Options** / **TCK Frequency**. See screenshot below:



Input the frequency you want to set. For example, if you want to change TCK frequency to 15MHz, you should input 15000000.
See screenshots below:



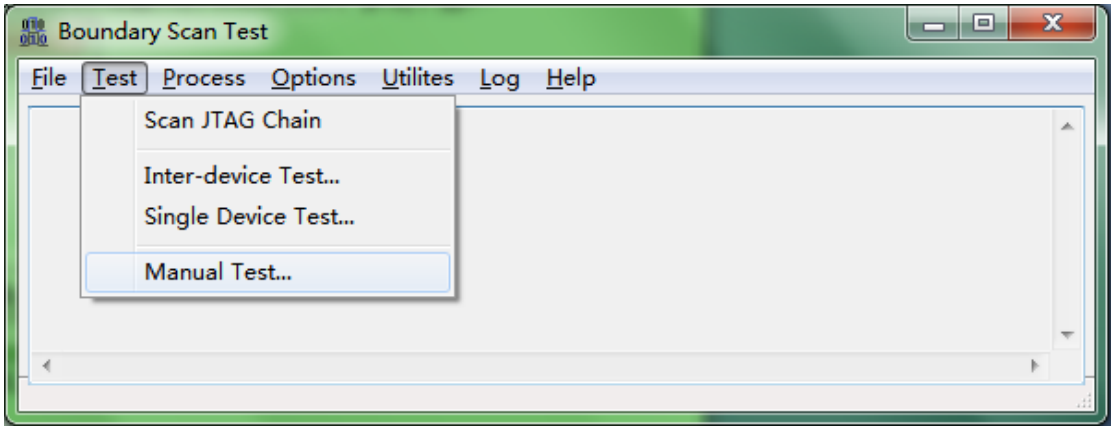
To WH-USB-HiJTAG cable, the frequency could be 30000000, 1500000, 10000000, and 6000000 and so on.
To WH-USB-JTAG cable, the frequency could be 6000000, 3000000 and so on.

Manual Test

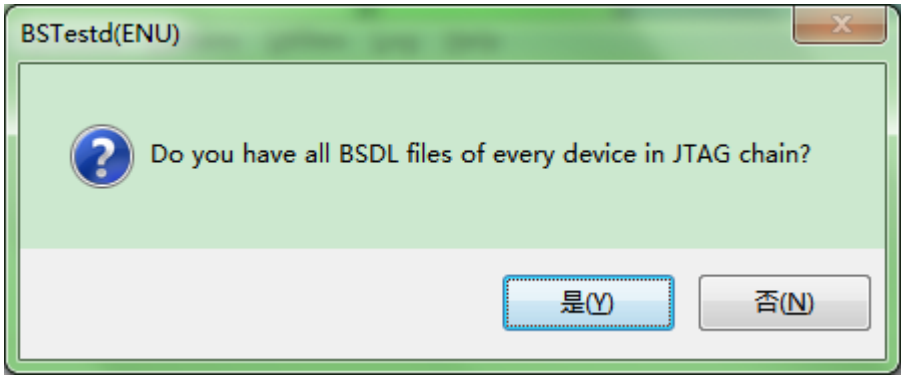
The system can display status of JTAG chip's pins (something like an oscilloscope or logic analyzer), and can control the chip to output a user defined waveform (something like a signal generator). You can find issues in welding (open, short, etc.) or PCB production by analyzing the phenomenon. Also, it can be used as debugging tool.

Preparation

Select menu **Test** - **Manual Test**. See screenshot below:

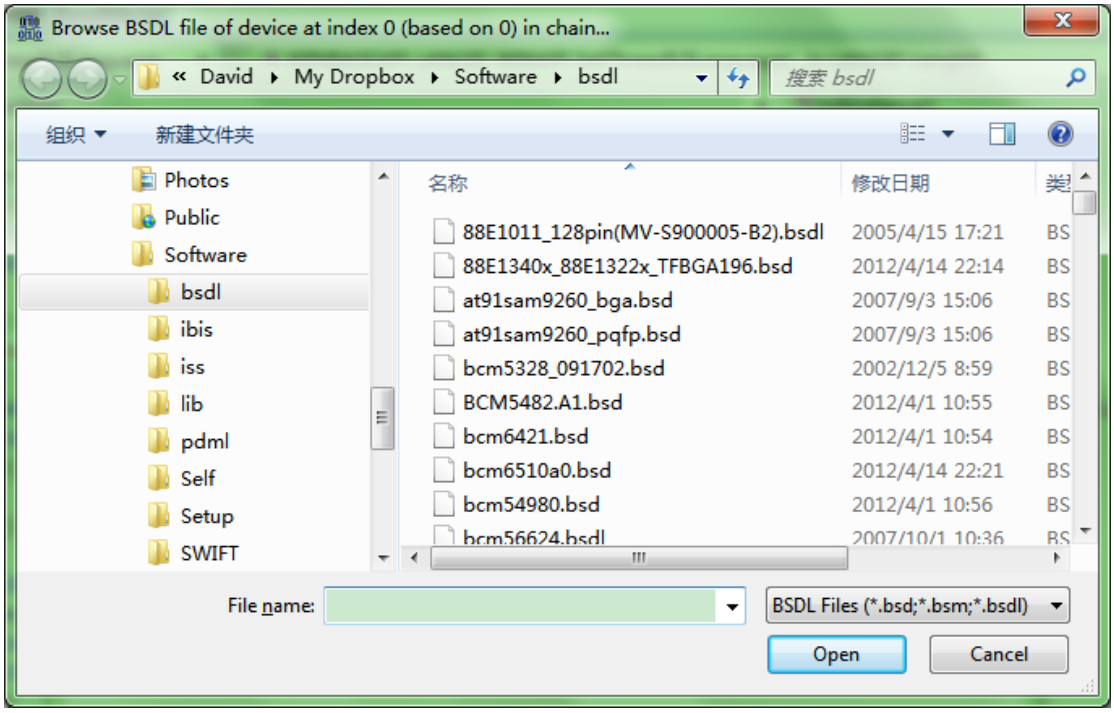


You will be asked whether you have BSDL file for every device in JTAG chain. See screenshot below:

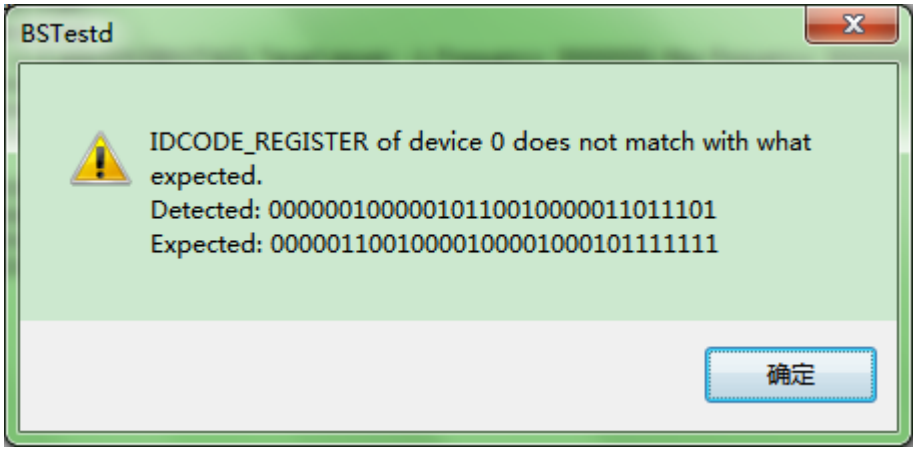


If you have BSDL file for every device

The software will let you choose BSDL file of device one by one. See screenshot below:



The software will parse each device's BSDL, and compare IDCODE in BSDL with detected IDCODE. If mismatch found, a message box is popped like this:

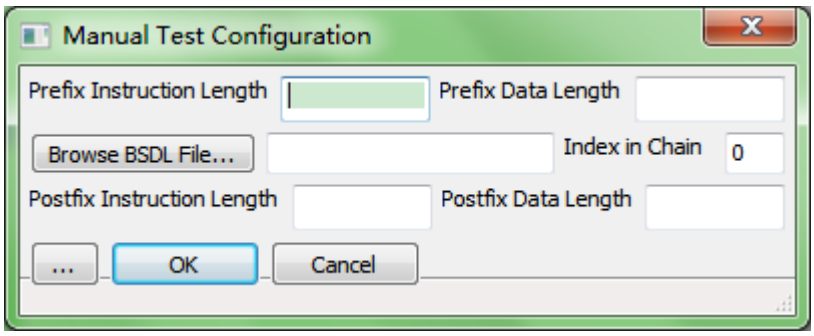


Note: Some JTAG device doesn't implement IDCODE instruction, and no IDCODE register in BSDL. So, the software could not judge whether BSDL file and device is matched or not. It's just a reminder.

If check passed, you will come to **Manual Test** dialog.

If you don't get BSDL file for every device

You will see a configuration dialog:



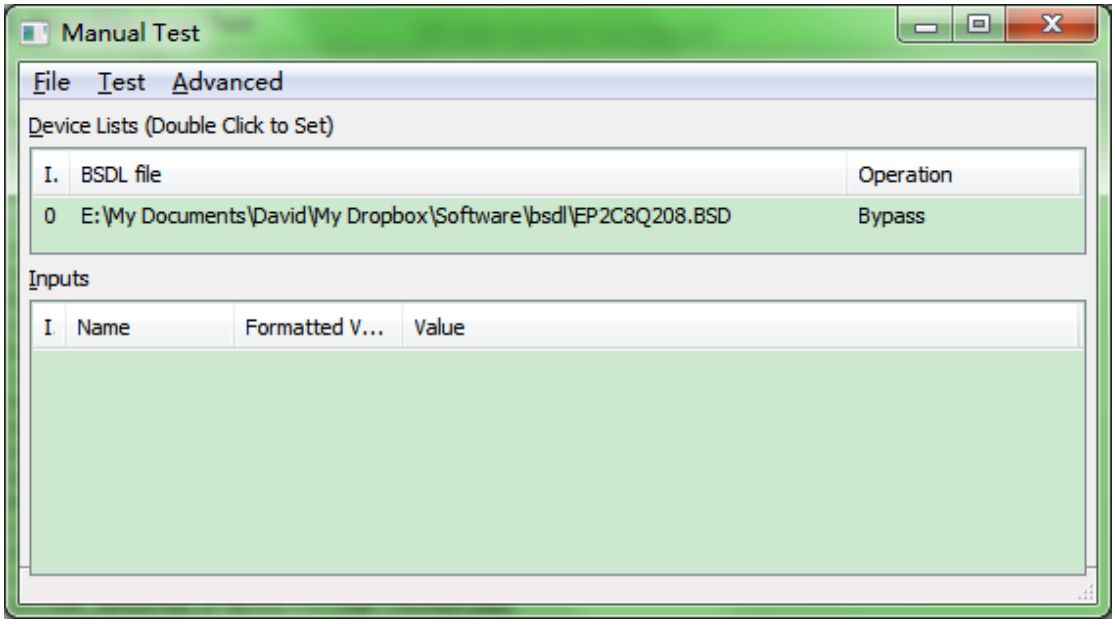
Prefix Instruction Length

Please input JTAG instruction length of all other devices ahead of DUT. If there are more than one devices, please split them with '|' character. For example, '4|5' means there are two devices ahead of DUT in JTAG chain, and device at index 0 has a 4-bit JTAG instruction while device at index 1 has a 5-bit JTAG instruction. And you could find that the DUT is at index 2.

Post Instruction Length

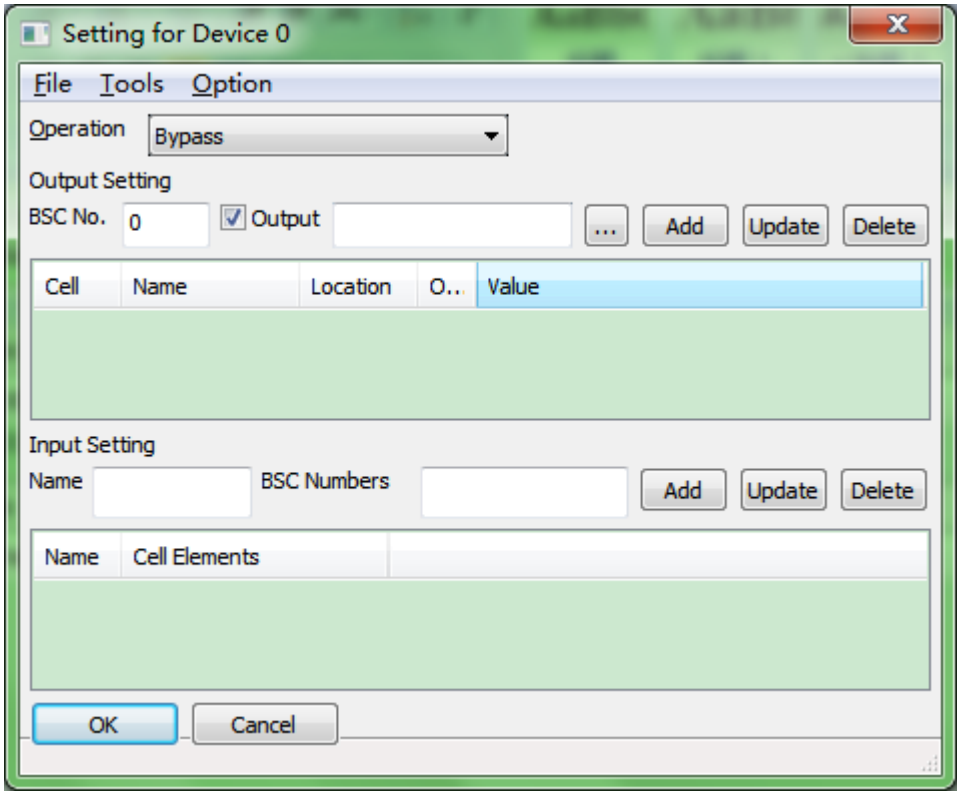
Please input JTAG instruction length of all other devices behind of DUT. Refer to Prefix Instruction Length.

Setting



By default the software will set operation to **Bypass**.

- Double click DUT in **Device Lists** to edit settings. You will see **Setting for Device n** dialog (n means index in JTAG chain and index begins with 0).



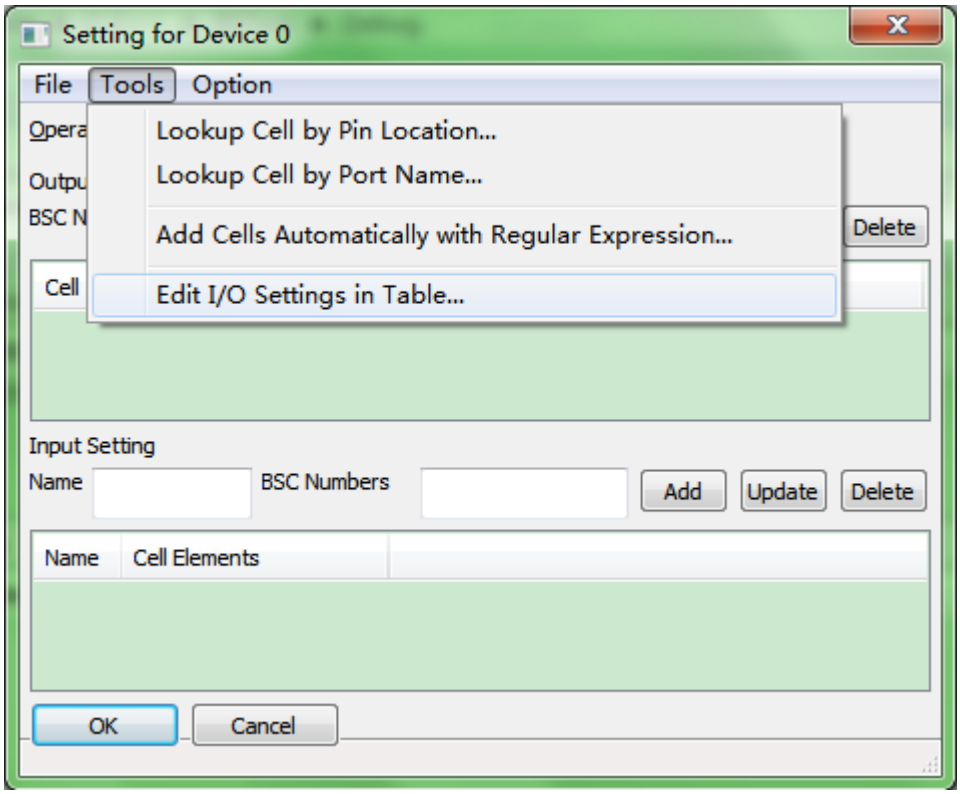
Operation

Change **Operation** to **Test** (for those devices that will not be tested, please keep '**Bypass**').

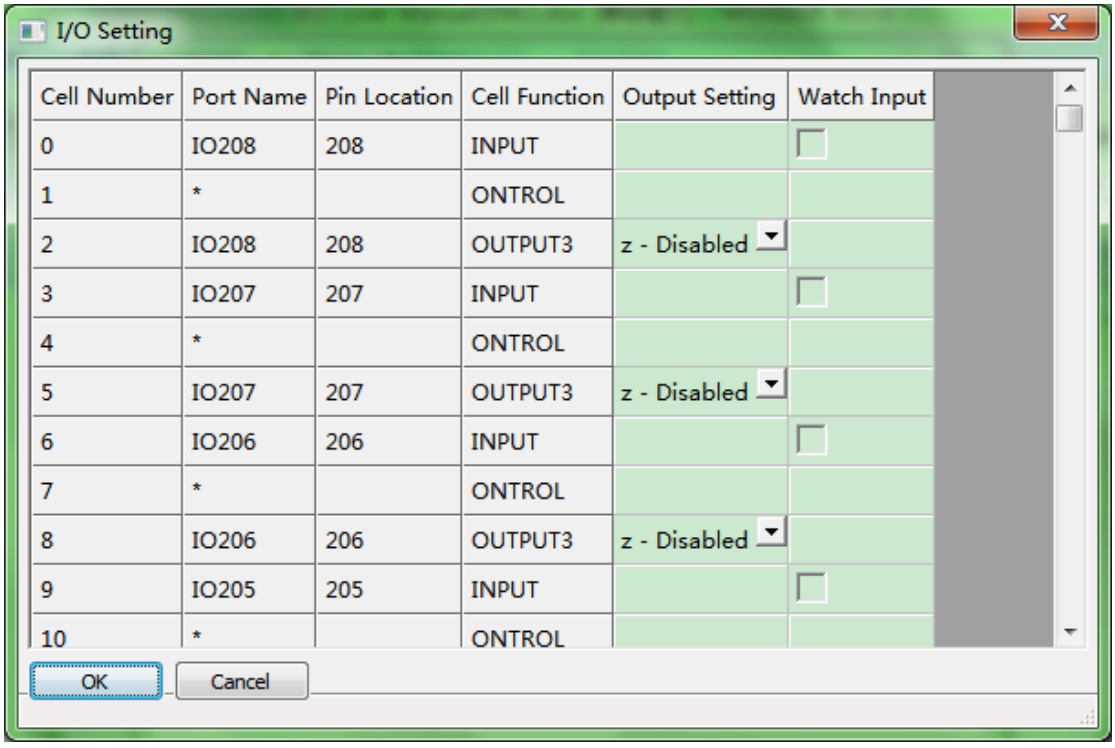
I/O

Utility for Setting I/O

Select menu **Utilities** / **Edit I/O Settings in Table**... . See screenshot below:



You will see **I/O Setting** dialog popped:



If you want to control a pin to output, please choose output value in table cell.

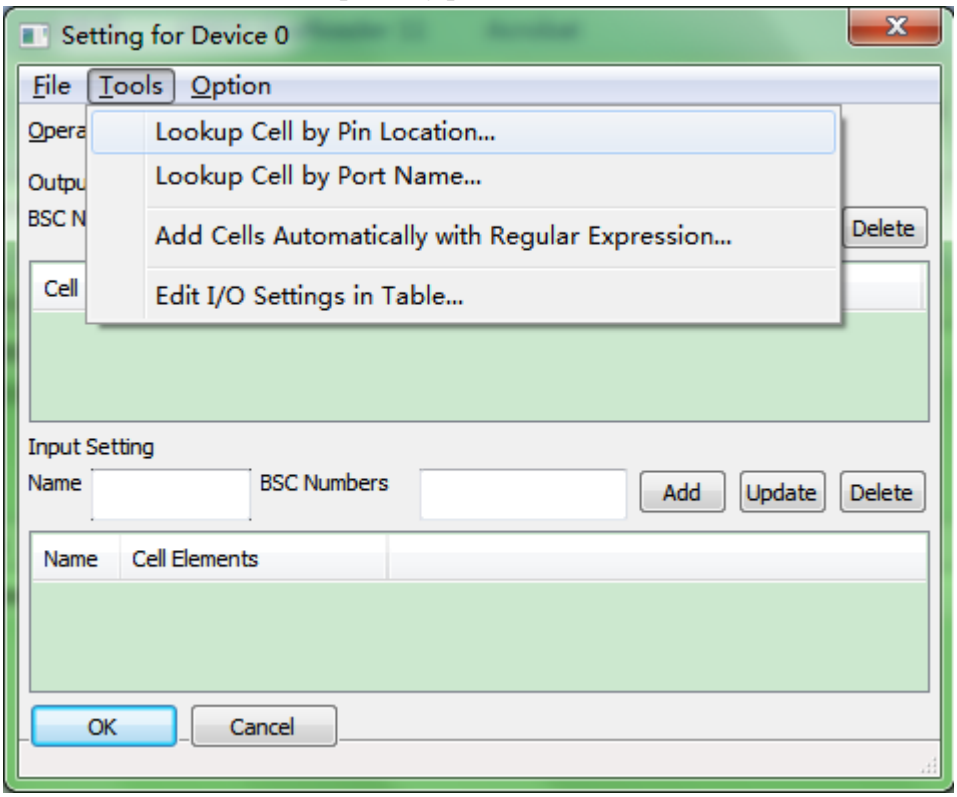
If you want to observe input value of a pin, please tick the checkbox.

When finished, please click **OK** to return **to Setting for Device n** dialog.

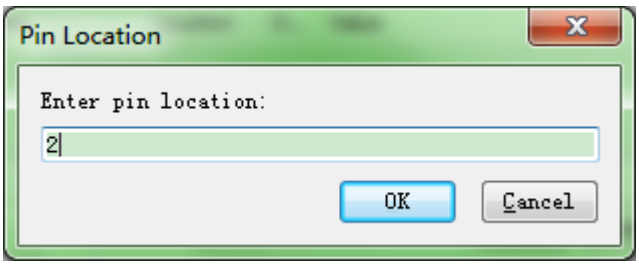
Utilities

Lookup BSC No. by Pin Location

Select menu **Utilities** –**Lookup cell by pin location...**:

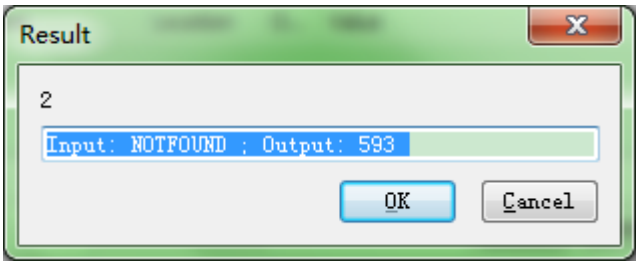


Input pin location. See screenshot below:



Click **OK** button.

The software will lookup BSDL file for the cell, and display search result.

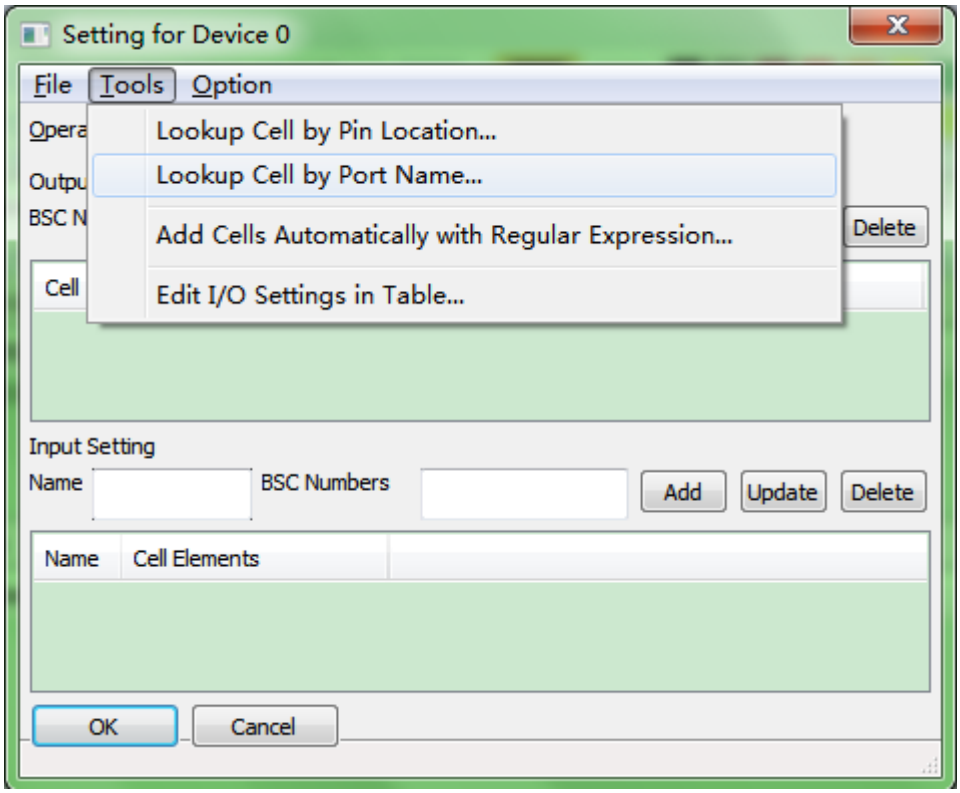


Now, you could copy corresponding cell number to **BSC No.** field for input or output

setting.

Lookup BSC No. by Port Name

Similarly, you could lookup a cell by port name.
Select menu **Utilities** –**Lookup cell by port name...** .



Output Setting

Input one in edit box behind **BSC No.** then input value (‘0’ or ‘1’) in edit box after **Output**.

Note: In fact, the software could output any pattern. That’s to say, you could input a binary string in the Output edit box. For example, if you set Output to 01, the software will output a 1:1 square waveform. If the Output pattern is 1000, the duty-cycle of output waveform will be 1:3.

Do above steps repeatedly until you have set all outputs.

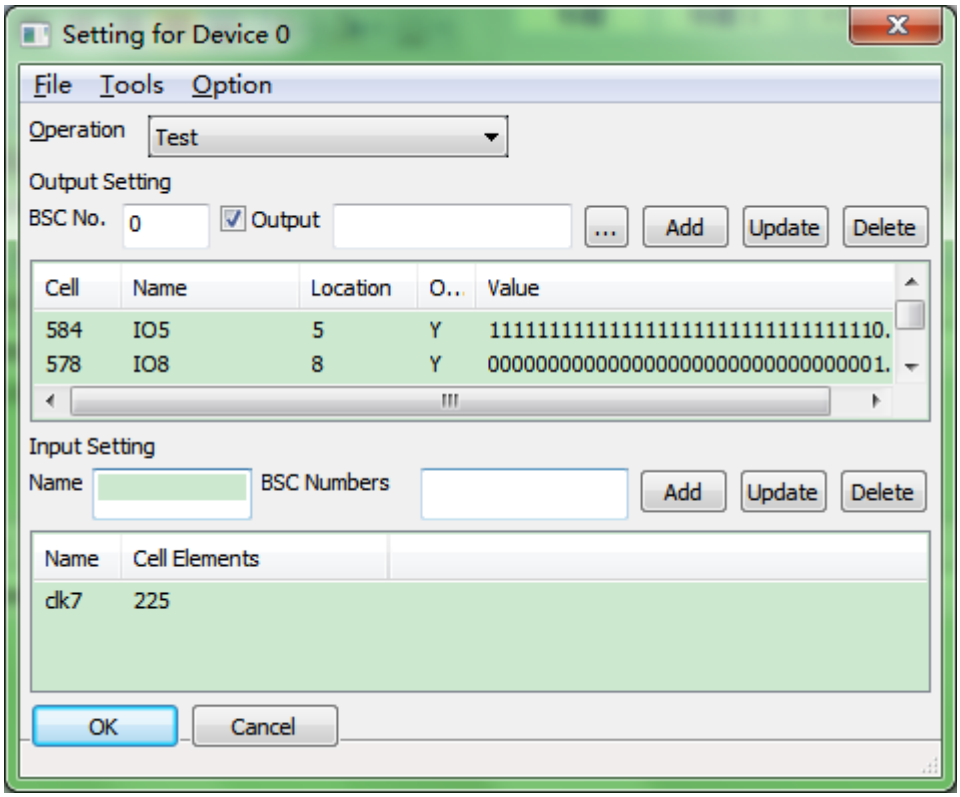
Input Setting

Input a name in edit box after ‘Name’, and input a cell number in edit box after **BSC number**, and click **Add** button.

Note: You can watch many input pins together and combine them in a group. Just input multi cell numbers and split them with ‘,’.

Do above steps repeatedly until you have set all inputs.

Here is an example:

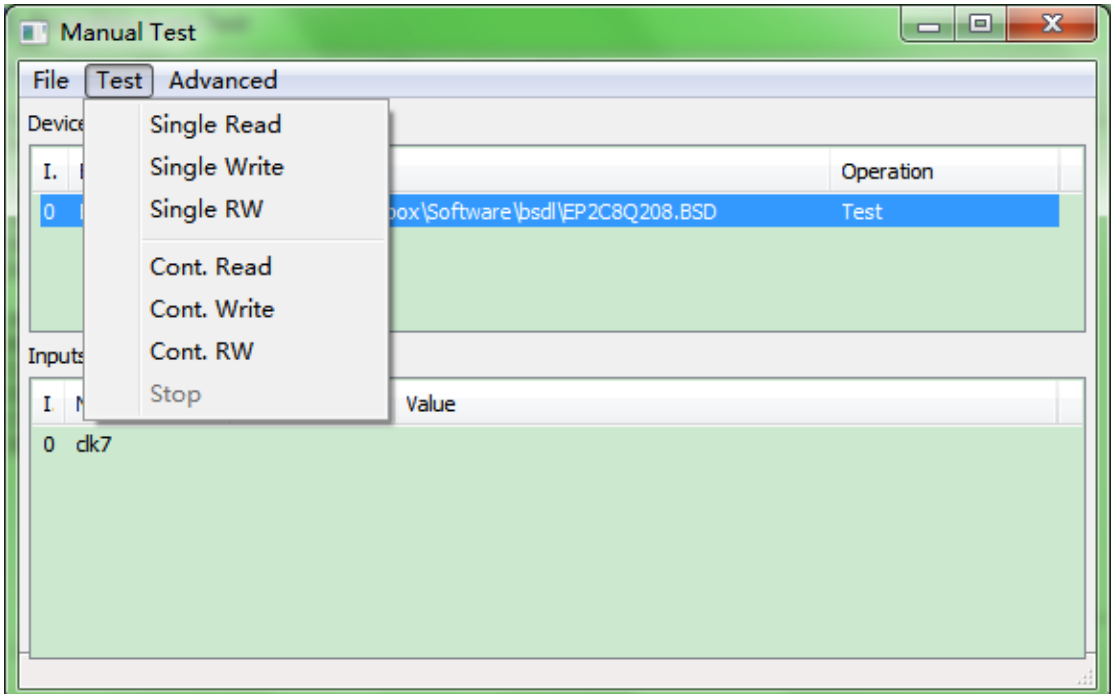


More Devices...

- Click **OK** button to return to **Manual Test** dialog.
- Do above steps repeatedly until you have set all devices.

Test

Test submenus:



Single Read: The software will read input value to pins one time of every device whose operation is in **Test**.

Single Write: The software will drive output value to pins one time of every device whose operation is in **Test**.

Single RW: **Single Write** and **Single Read**.

Cont. Read: The software will loop doing **Single Read** until menu **Test – Stop** is clicked.

Cont. Write: The software will loop doing **Single Write** until menu **Test – Stop** is clicked.

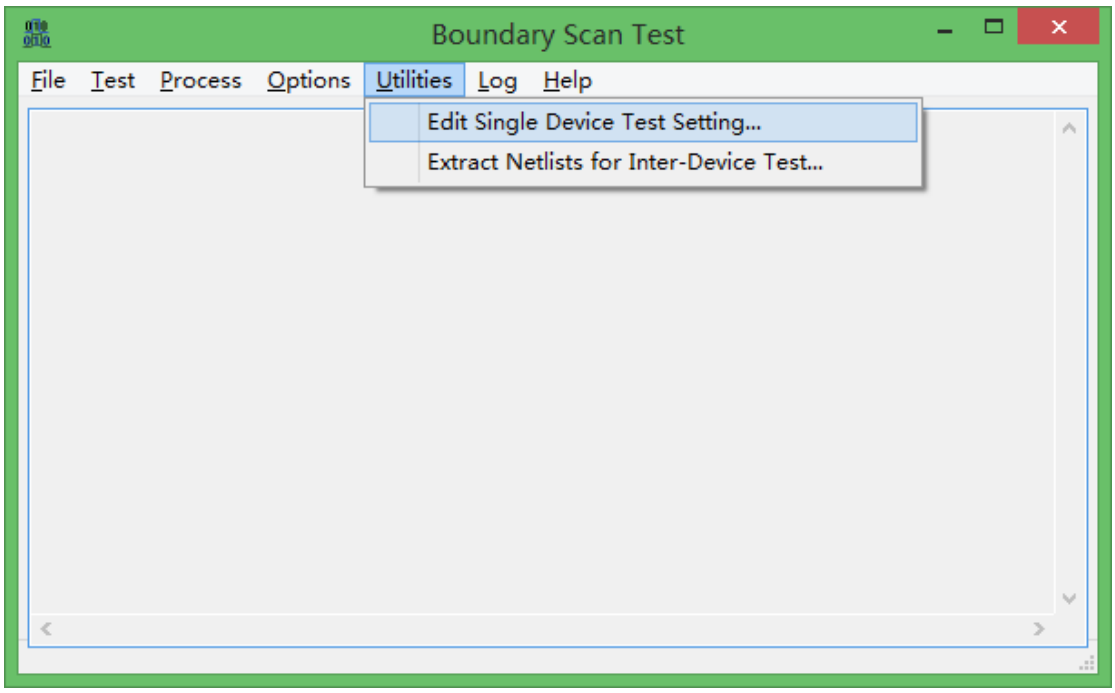
Cont. RW: The software will loop doing **Single RW** until menu **Test – Stop** is clicked.

Single Device Test

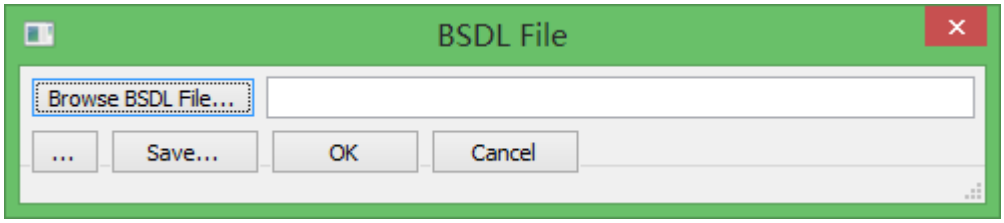
This automatic operation can detect short between pins (in fact, not only pins of JTAG device, but also pins of other devices connected to JTAG device), or short between I/O pins and power supply or ground.

Preparation

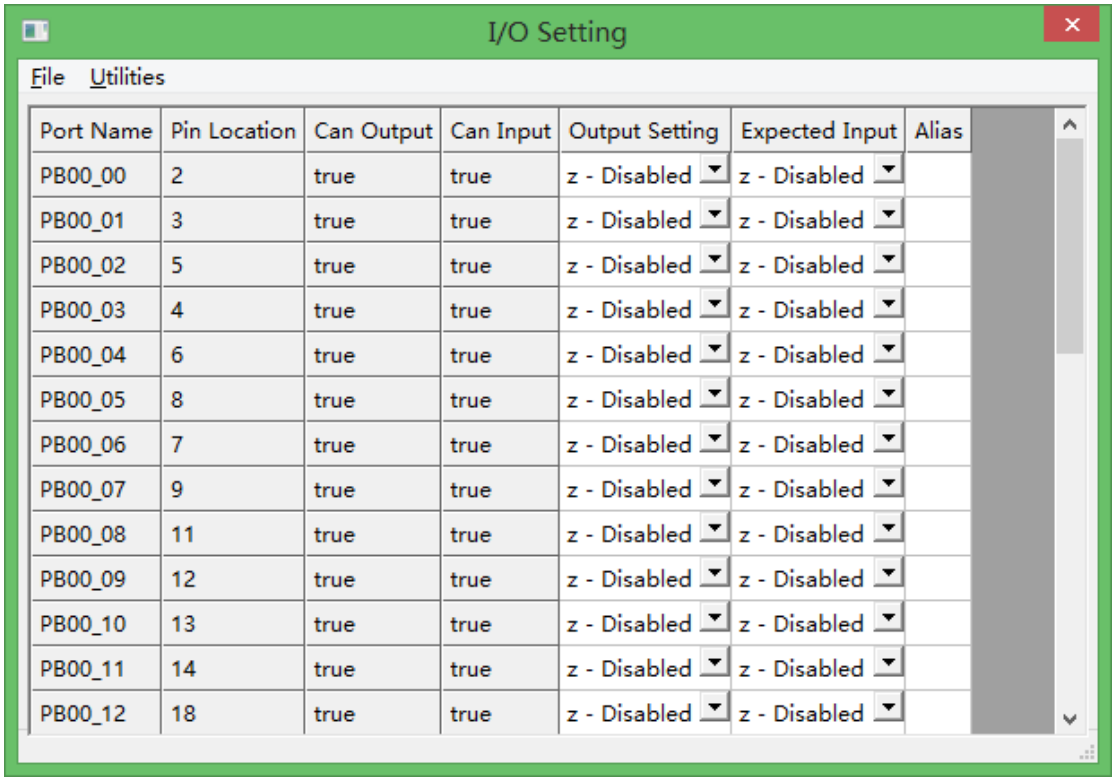
Before testing, please create a setting file by selecting menu **Utilities** –**Edit Single Device Test Setting…**:



A '**BSDL File**' dialog will popup. Select correct BSDL and click **OK**:

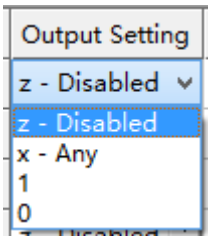


Now the '**I/O Setting**' dialog is shown:



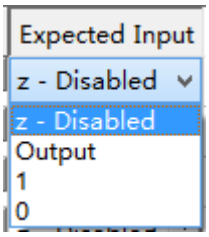
Port Name	Pin Location	Can Output	Can Input	Output Setting	Expected Input	Alias
PB00_00	2	true	true	z - Disabled	z - Disabled	
PB00_01	3	true	true	z - Disabled	z - Disabled	
PB00_02	5	true	true	z - Disabled	z - Disabled	
PB00_03	4	true	true	z - Disabled	z - Disabled	
PB00_04	6	true	true	z - Disabled	z - Disabled	
PB00_05	8	true	true	z - Disabled	z - Disabled	
PB00_06	7	true	true	z - Disabled	z - Disabled	
PB00_07	9	true	true	z - Disabled	z - Disabled	
PB00_08	11	true	true	z - Disabled	z - Disabled	
PB00_09	12	true	true	z - Disabled	z - Disabled	
PB00_10	13	true	true	z - Disabled	z - Disabled	
PB00_11	14	true	true	z - Disabled	z - Disabled	
PB00_12	18	true	true	z - Disabled	z - Disabled	

Output Setting



Option	Output Status	Comments
z - N/A	This cell could not output.	
z – Disabled	This cell is outputable, but we don’t test it.	
x – Any	Output anything.	*Pin under test should select this option.*
1	Output ‘1’ always.	The pin will stay on a certain status. It could be used to control some circuit.
0	Output ‘0’ always.	The pin will stay on a certain status. It could be used to control some circuit.

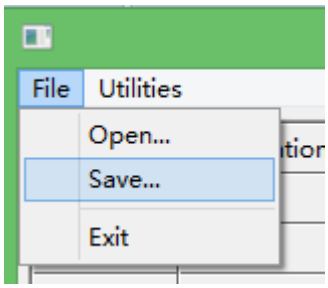
Expected Input



Option	Input Status	Comments
z - N/A	This cell could not input.	
z - Disabled	This cell is inputable, but we don't check it.	
Output	Changes with the Output.	Input should be the same as output. *Pin under test should select this option.*
1	Input is '1'.	This pin should always input '1' normally, and other status when failure. It happens to the pin whose status is fixed.
0	Input is '0'.	This pin should always input '0' normally, and other status when failure. It happens to the pin whose status is fixed.

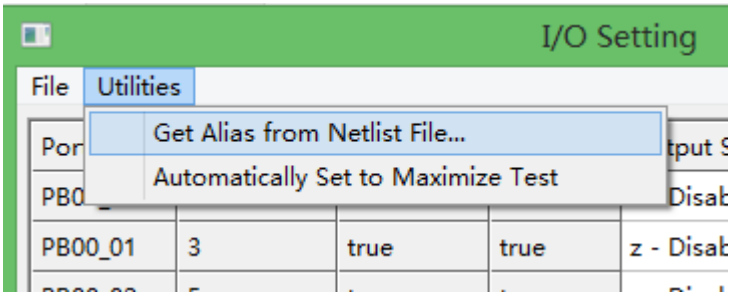
Save

Menu **File** – **Save…**:



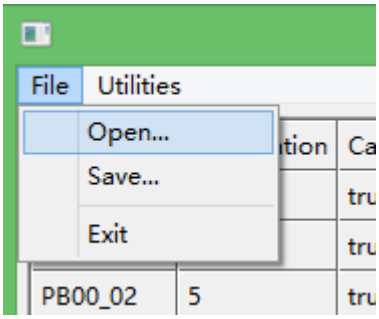
Utilities

To improve the readability of test result, you could add alias to the pins. Select menu **Utilities - Get Alias from Netlist File…**:



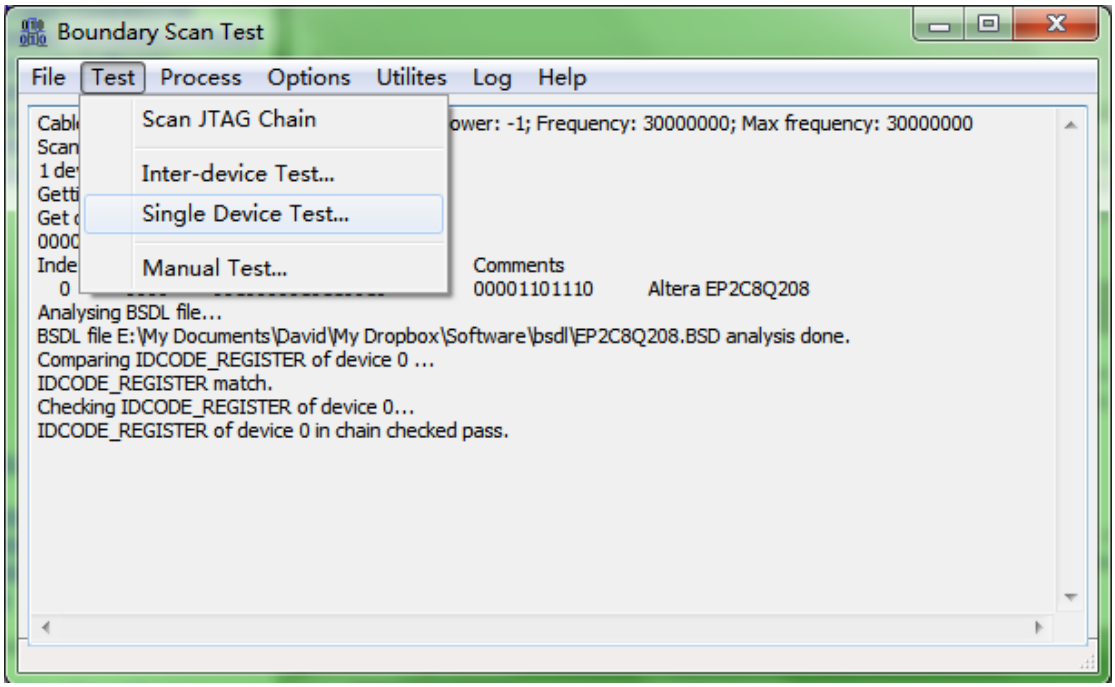
Edit Saved Setting File

Menu **File – Open…**

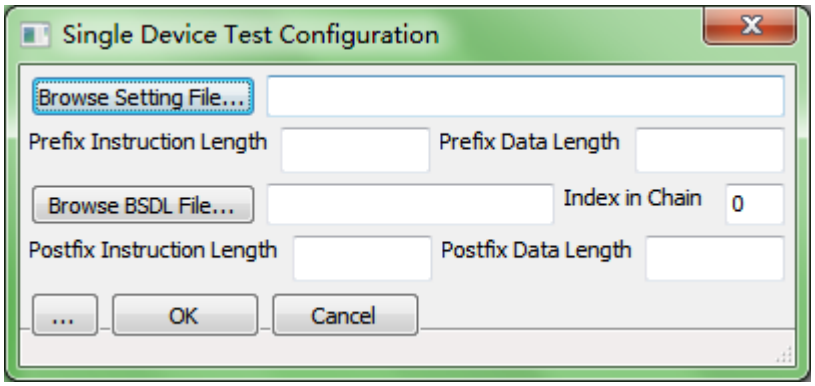


Steps

Select menu **Test** - **Single Device Test**:



You will see **Single Device Test Configuration** dialog:



Browse Setting File: Select the saved setting file in above section.

Prefix Instruction Length: Please input JTAG instruction length of all other devices ahead of DUT. If there are more than one devices, please split them with '|' character. For example, '4 | 5' means there are two devices ahead of DUT in JTAG chain, and device at index 0 has a 4-bit JTAG instruction while device at index 1 has a 5-bit JTAG instruction. And you could find that the DUT is at index 2.

Post Instruction Length: Please input JTAG instruction length of all other devices behind of DUT. Refer to Prefix Instruction Length.

When **Setting File**, **BSDL file of DUT** and **Index of DUT** are set, click **OK** button.

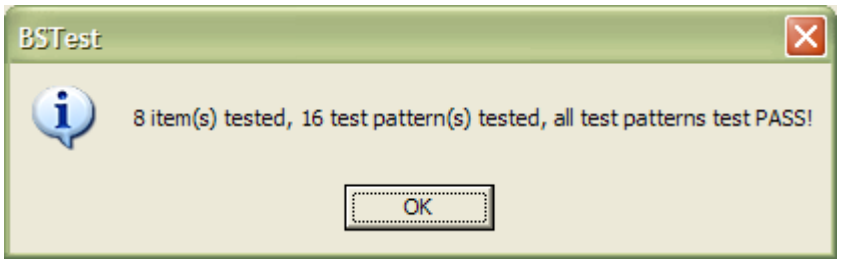
The software will run rest according to **Setting File** and show testing result.

Example 1

The **Setting File** contains 8 items to be tested. They are `D0` to `D7`.

Test on Good Board

No error found.

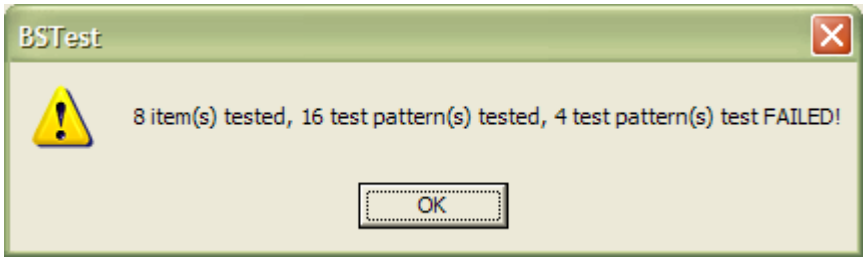


The log will show detailed info of test.

```
Single Device Test is running...
Output 0  Input     Result    Port
01111111 01111111  G6 / D0
10111111 10111111  H9 / D1
11011111 11011111  H7 / D2
11101111 11101111  H4 / D3
11110111 11110111  G3 / D4
11111011 11111011  H6 / D5
11111101 11111101  H1 / D6
11111110 11111110  I2 / D7
Output 1  Input     Result    Port
10000000 10000000  G6 / D0
01000000 01000000  H9 / D1
00100000 00100000  H7 / D2
00010000 00010000  H4 / D3
00001000 00001000  G3 / D4
00000100 00000100  H6 / D5
00000010 00000010  H1 / D6
00000001 00000001  I2 / D7
8 item(s) tested, 16 test pattern(s) tested, all test patterns test PASS!
```

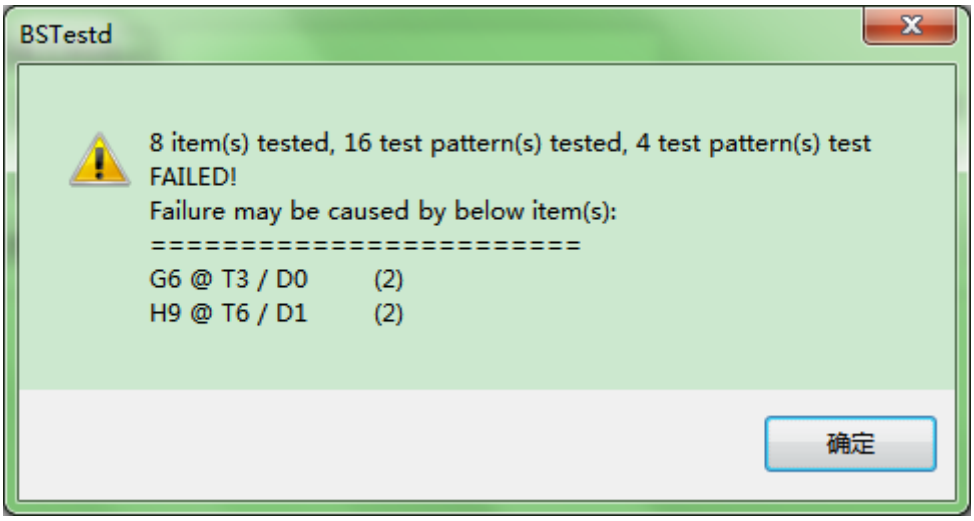
Short-Circuit Issue

We manually connect pin `D0` and `D1` with a wire. This time we'll be told failure found in test.



The software will tell you failure items detected since V2.2.2.1. Furthermore, the software

will show pin location of failed item since V2.3.0.0. Display format: **Port Name @ Pin Location / Alias**. Please note pin location or alias will not be displayed is it's empty. See screenshot below:


BSTestd
8 item(s) tested, 16 test pattern(s) tested, 4 test pattern(s) test FAILED!
Failure may be caused by below item(s):
==========
G6 @ T3 / D0 (2)
H9 @ T6 / D1 (2)
确定

Log showed:

```
Single Device Test is running...
Output 0  Input     Result     Port
01111111 11111111 ? G6 / D0
10111111 11111111 ? H9 / D1
11011111 11011111  H7 / D2
11101111 11101111  H4 / D3
11110111 11110111  G3 / D4
11111011 11111011  H6 / D5
11111101 11111101  H1 / D6
11111110 11111110  I2 / D7
Output 1  Input     Result     Port
10000000 11000000 ? G6 / D0
01000000 11000000 ? H9 / D1
00100000 00100000  H7 / D2
00010000 00010000  H4 / D3
00001000 00001000  G3 / D4
00000100 00000100  H6 / D5
00000010 00000010  H1 / D6
00000001 00000001  I2 / D7
8 item(s) tested, 16 test pattern(s) tested, 4 test pattern(s) test FAILED!
```

It's easy to find `D0` and `D1` didn't pass the test since we shorted them.

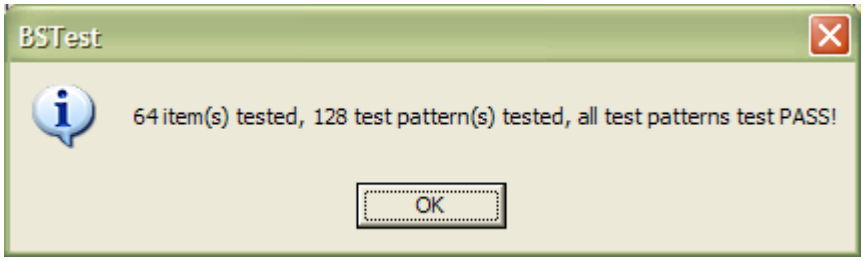
More...

In fact, not only short circuit between pins of JTAG device, but also short circuit between pins of other devices connected to JTAG device could be found.

Example 2

The **Setting File** contains 64 items to be tested. They are `d(0)` to `d(63)`.

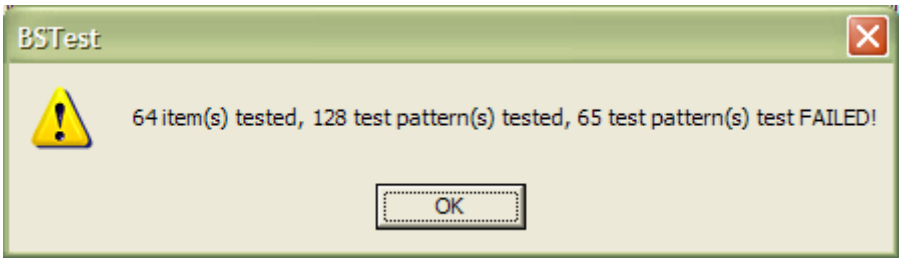
Test on Good Board



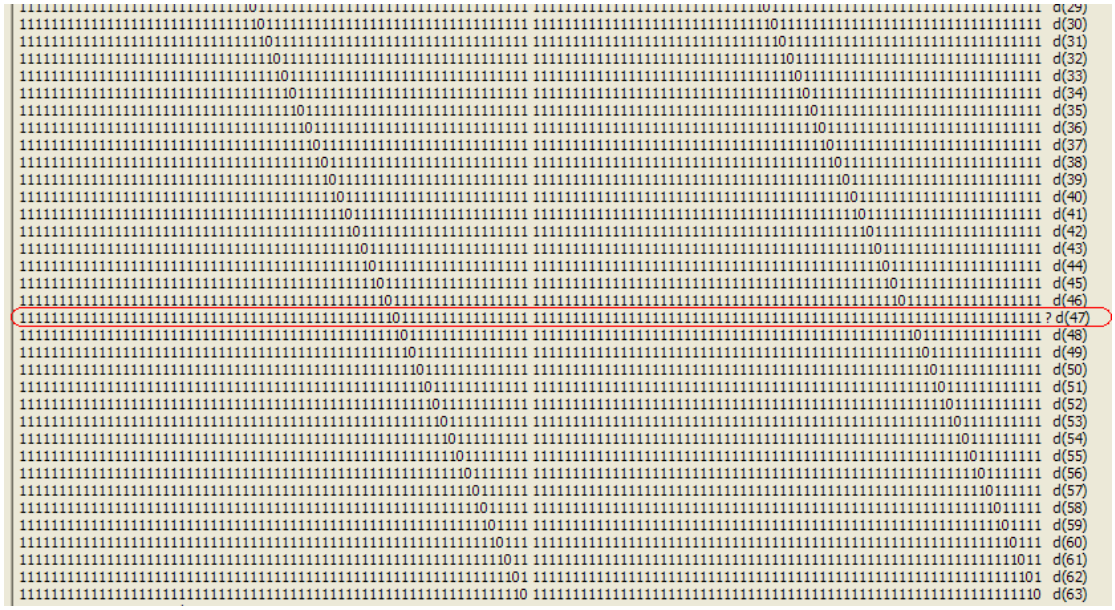
No error found.

Short Circuit with Power

We manually connect pin `d(47)` and `3.3V` with a wire.



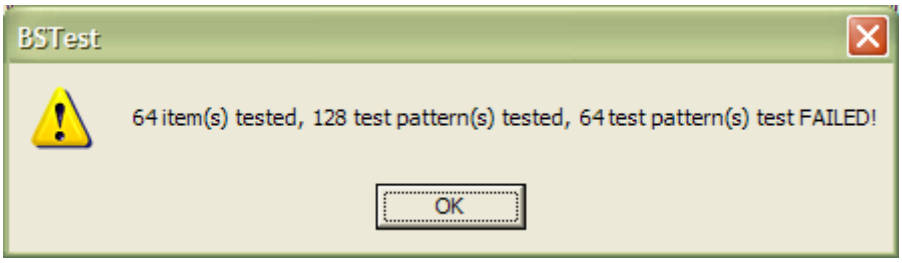
Detailed log:



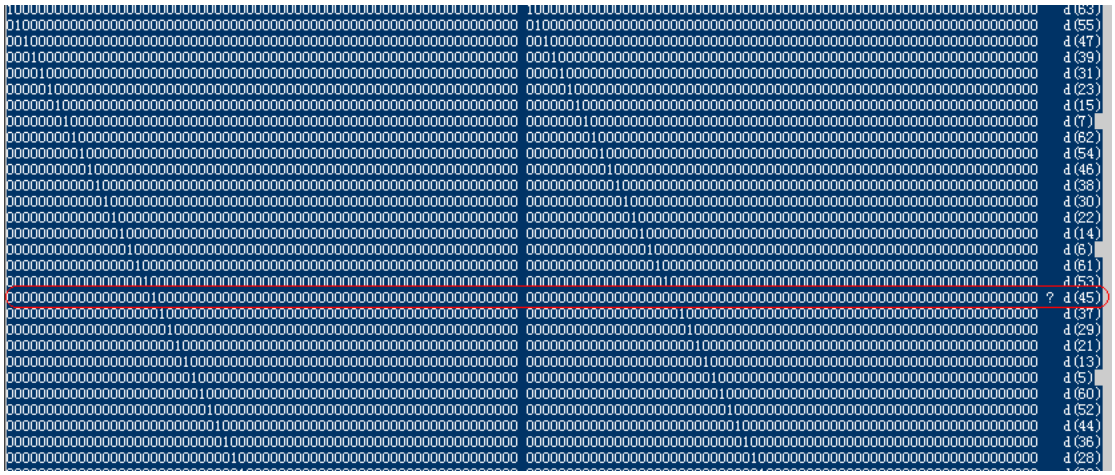
We could see easily that pin `d(47)` doesn't pass the test.

Short Circuit with GND

We manually connect pin `d(45)` and `GND` with a wire



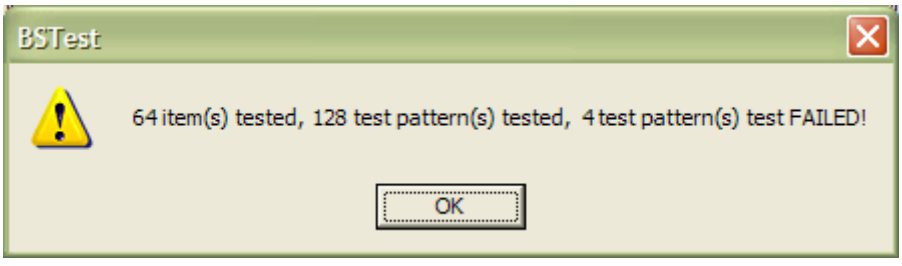
Detailed log:



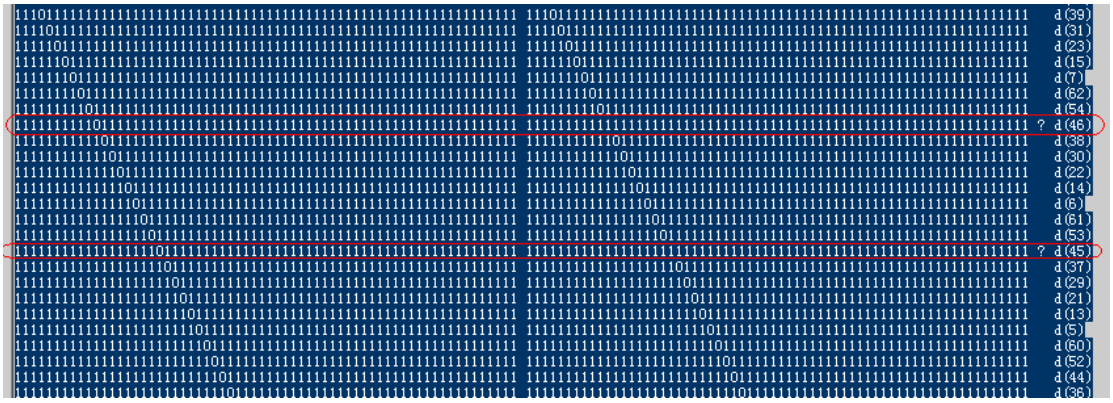
Obviously, `d(45)` could not pass the test.

Short-Circuit Between Pins

We manually connect pin `d(45)` and `d(46)` with a wire.:



Detailed log of test when outputting 0:



Detailed log of test when outputting 1:

We could see that there are always two pins test failed no matter what value is outputting (0 or 1). The two pins are `d(45)` and `d(46)`.

More...

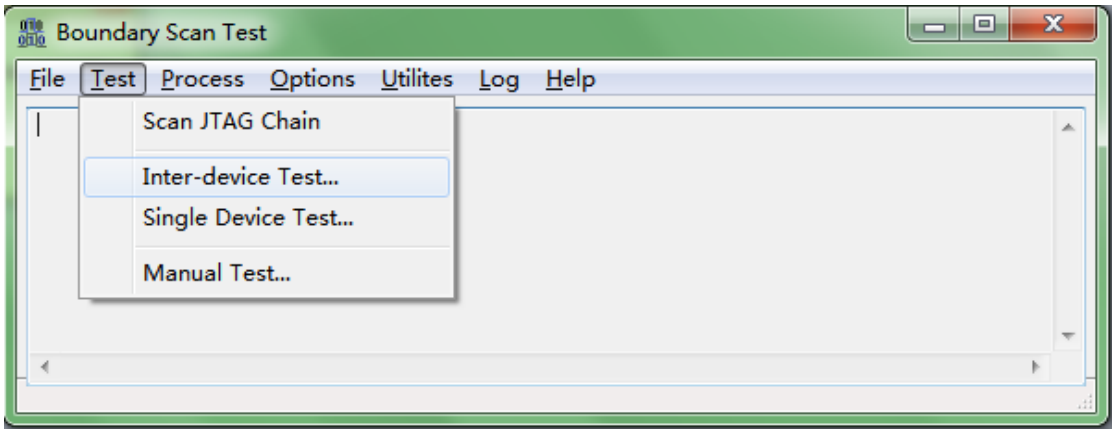
Examples above just list part of pins. You can add as more as you want in the test setting file.

Inter-Device Test

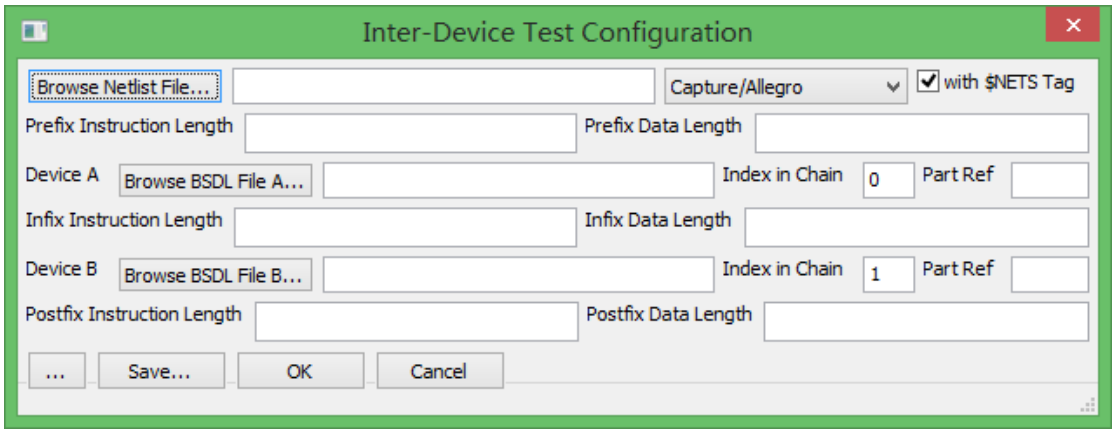
The system reads schematic netlists, and analyzes the device inter-connection, then generates test patterns. Inter-device test can automatically find short (pin to pin, pin to power supply or ground), open and other issues.

Steps

Select menu **Test** – I**nter-device Test**. See screenshot below:



You will see **Inter-device Test Configuration** dialog:

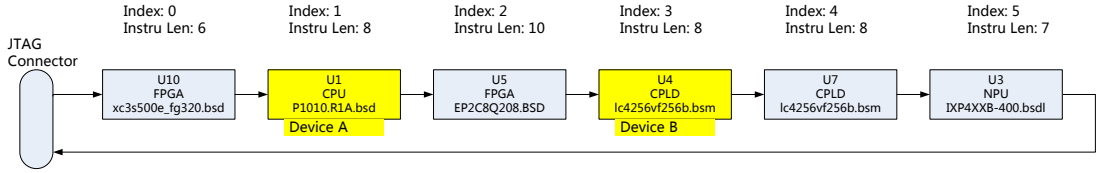


About the Parameters

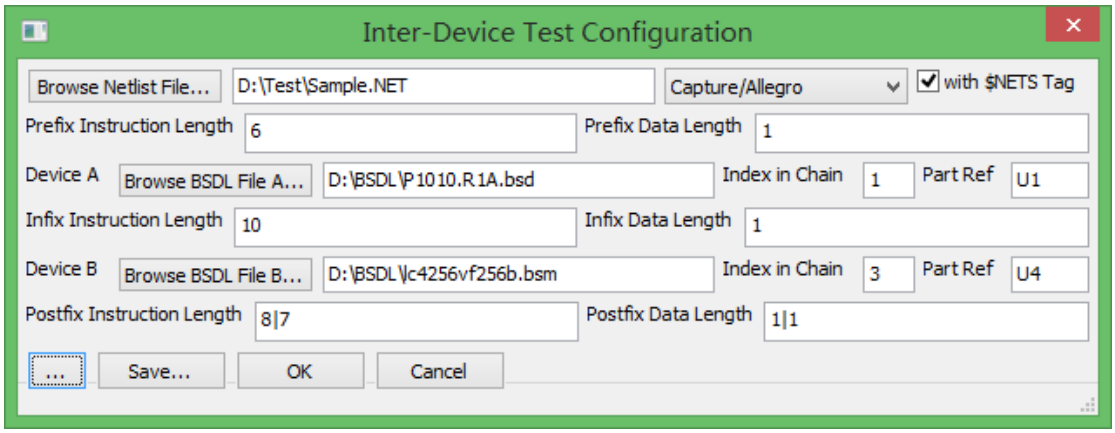
There are two DUTs. Let's call them DUT A / **Device A** and DUT B / **Device B**.
Set schematic netlist file, and choose the netlist file format.
Set **BSDL file**, **Part Ref** and **Index in JTAG Chain** for DUT A and B.
Prefix Instruction Length: Please input JTAG instruction length of all other devices ahead of DUT A. If there are more than one devices, please split them with '|' character. For example, '4 | 5' means there are two devices ahead of DUT in JTAG chain, and device at index 0 has a 4-bit JTAG instruction while device at index 1 has a 5-bit JTAG instruction. And you could find that the DUT is at index 2. **Must left blank if no such device.**
Infix Instruction Length: Please input JTAG instruction length of all other devices between DUT A and DUT B. Refer to Prefix Instruction Length. **Must left blank if no such device.**
Postfix Instruction Length: Please input JTAG instruction length of all other devices behind of DUT B. Refer to Prefix Instruction Length. **Must left blank if no such device.**

An Example

Let's suppose a board with 6 devices in a single JTAG chain, and we are going to do inter-device test of U1 and U4.



The setting should be:



Notes:

- All indexes are zero based.
- Except U1 and U4, other device are Bypass. The data length of the Bypass device is 1, so the data length is not required.

Tips

For you convenience, you could click **Save…** button to save the parameters to a text file with extension .ini. Next time just click the button to load saved parameters.

Run the Test

Click **OK** after all parameters are set.

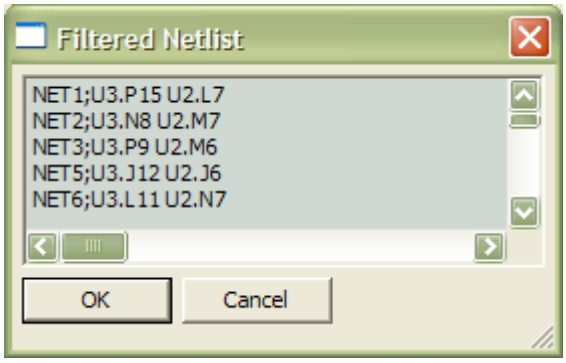
The software will do a JTAG chain scan and show result in log.

Then the software will parse BSDL files of devices in chain.

Then the software will check IDCODE.

After IDCODE checking passed, the software will parse netlist file.

The software will filter connection from all netlist according to Part Ref. You will see **Filtered Netlist** dialog.

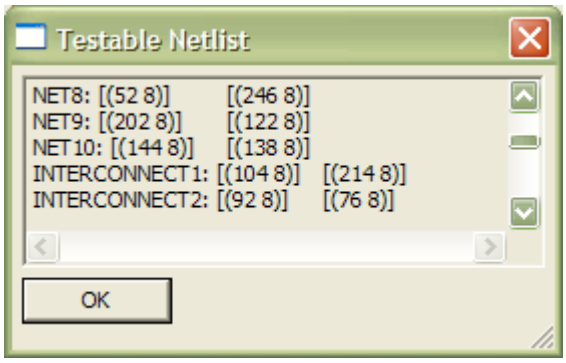


It's a simple editor. You can edit the connections if you want.
That's because sometimes the pins are connected through a resistor or logic clue, but they will not be filtered automatically since they have same netlist name. But they are testable. You can add them manually.

Click **OK** to continue.

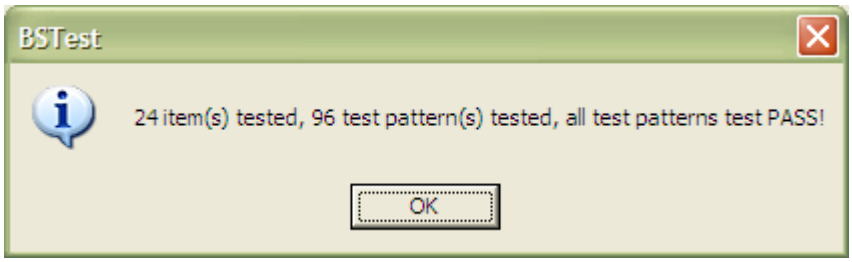
The software will analyze to find testable netlists.
You know, only netlist connected between DUT A and DUT B could be tested, and those connections that one device could output and the other device could input could be testable.

Testable Netlist dialog will show testable netlists.



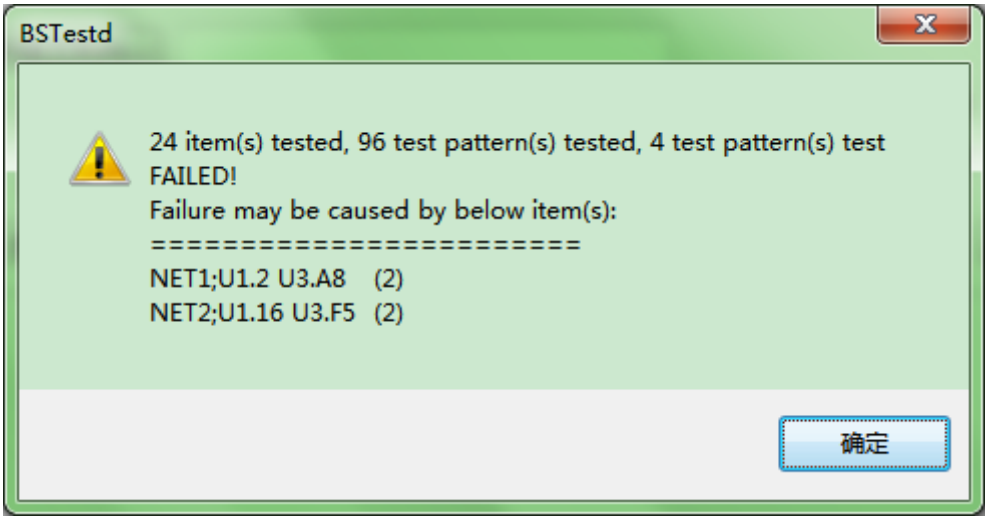
Click **OK** button.

The software will run test and show result.



You could check detailed info in log.

The software will tell you failure items detected since V2.2.2.1.
The software will show pin location at the same time since V2.3.0.0.
An example screenshot:


BSTestd
24 item(s) tested, 96 test pattern(s) tested, 4 test pattern(s) test FAILED!
Failure may be caused by below item(s):
==========
NET1;U1.2 U3.A8 (2)
NET2;U1.16 U3.F5 (2)
确定

UI Difference Under Different Launching Mode

UI \ Launching Display	Menu: Inter-Device Test	Menu: Batch Test	Automatically launched when startup
Display 'Edit Netlists' Dialog	Yes	No	No
Display 'Testable Netlists' Confirmation Dialog	Yes	No	No
Show Error Message	Yes	Yes	Yes
Show Success Message	Yes	Yes	No

PLD Configuration

Currently .svf and .vme files are supported.
Note:

- *A PLD maybe a CPLD/EPLD, and could also be a FPGA.*
- *.svf file could be used to perform a test besides configuration.*

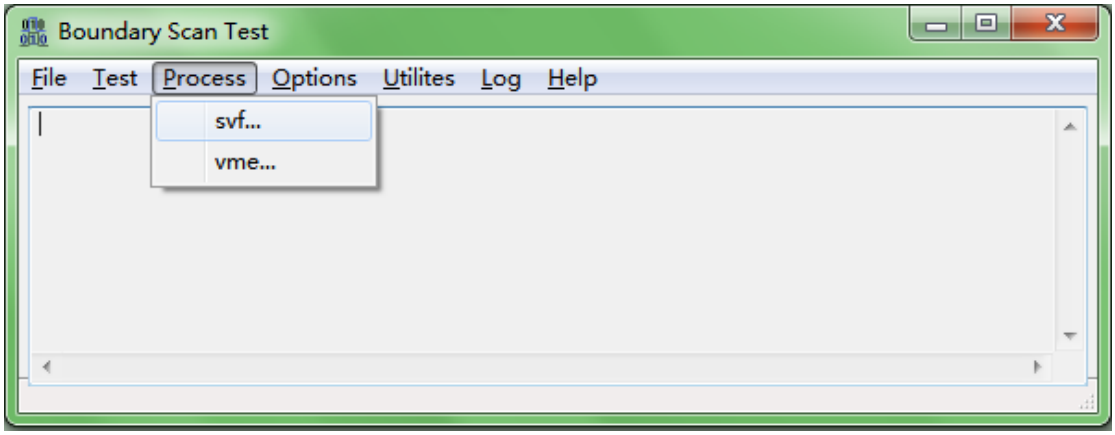
We show example of .svf in this manual. To .vme file, it's very similar.

Get .svf File

Almost every EAD software could convert programming file from other format to .svf format. Ask help from your PLD vendor if you have difficulty.

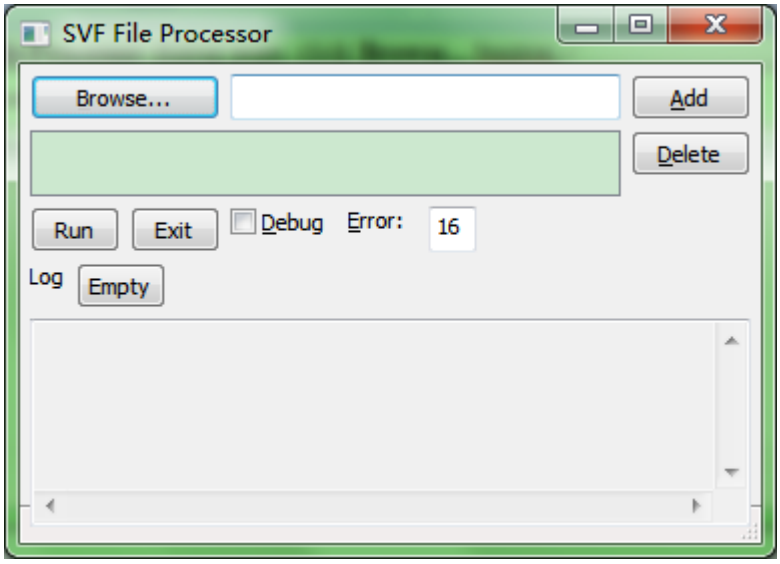
Start

Select menu **Process** - **.svf File**. See screenshot below:

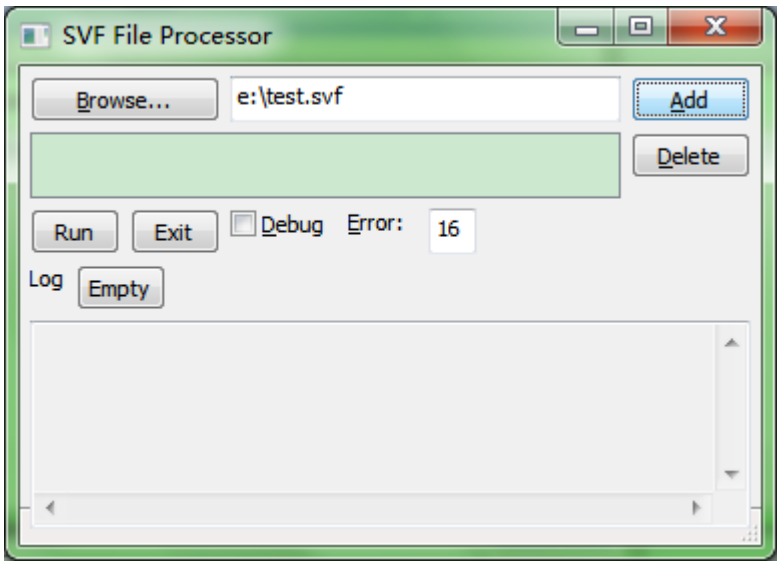


Set .svf File

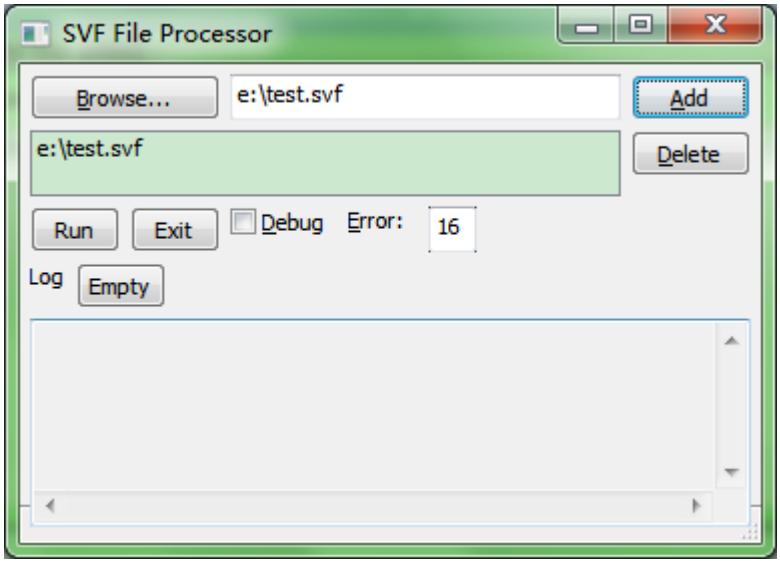
When **SVF File Processor** dialog pops, click **Browse…** button.



Click **Add** button after file is selected.

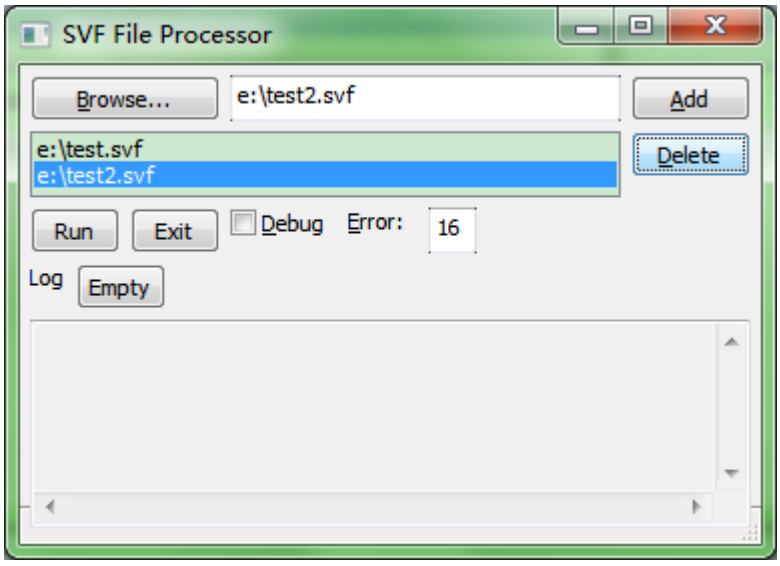


Screenshot after file added:



You may add multi files. The software will process them one by one.

If you want to remove one svf file, just select it and click **Delete** button. See screenshot below:



Run

Click **Run** button when setting is done.
The software will show the run result.

Revision History

Date	Version	Author	Changes
2014/12/2			● Move V2.2.2.1 improvement to first failure case; ● Add V2.3.0.0 feature;
2014/11/20			Add section UI Difference Under Different Launching Mode;
2014/11/14			Add screenshot of V2.2.2.1 which will show detected failure pins besides failure counter;
2014/11/13			● Add Preparation section in Single Device Test; ● Add An Example in Inter-Device Test; ● Add Tips in About the Parameters section in Inter-Device Test; ● Update text and screenshot because more netlist file formats are supported;
2014/6/9			Add line above footer;
2013/7/30			Format optimization;
2013/7/28			First Release